Risk Assessment

	Description	Evaluation	Likelihood	Impact Level	Responsibility	Response	Control Measures
1	Failing to complete full scope of project by deadline	The completed project will not be uploaded to GitHub before the deadline, leading to a fail	Low	High	Developers	Submit work in stages - uploading frequently to GitHub. Submit everything that had been done up until the deadline, noting in the README.md which elements are missing, and why.	Plan time at beginning of project complete with daily goals. Stick to daily goals and factor in extra time in case of unforseen problems. Complete MVP and upload to GitHub before adding additional features.
2	Broken version deployed onto production	Deployed application may have compile/runtime errors or miss key requirements	Low	High	Developers	Revert application to most recent stable version	Automate tests before deployment and making a pull request from feature/dev to main branch.
3	Injury/illness	Developer is compromised by illness or injury, making them unable to complete the project	Medium	Medium	Developers	In line with severity of illness/injury, speak to assessors to negotiate additional time or a reduction in scope of MVP.	Take frequent breaks, maintain good physical health through exercise, healthy eating and avoiding Covid
4	Computer system failure	Computer used for project suffers from a system failure or severe memory loss.	Low	High	Laptop Provider	Depending on severity of failure/memory loss, speak to assessors to negotiate additiona time/reduction in scope. Revert program to last saved version on GitHub.	Frequently back up computer to ensure the survival of application code and data. Use a VCS to ensure there is a remote backup which can be accessed from another computer if necessary.
5	VCS compromised	Remote repository is lost due to a hostile attack of GitHub	Low	High	GitHub	Re-upload project to alternate VCS and share with assessors.	Keep a local version of the application in case of the need to deploy to an alternative remote repo.
6	Plagiarism	Another classmate copies project from VCS and uses code as their own.	Low	High	QA	Assessors must pay attention to commit history on VCS to ascertain which student's work is their own, and which has been plagiarised.	Make code unique and embed choices that you would only be able to justify/understand as the author of the code.
7	Failure to include functional .jar file in repo	Assessors will not be able to run the application to check it works.	Medium	Medium	Developers	Revert application to most recent stable version with working .jar file.	Fully test .jar file before uploading it, ensuring it contains all dependencies needed by application